

# コラッツ予想マッピング ～マッピングというよりナビゲーション～

This study investigates the Collatz conjecture with using numbers that give a remainder of 4 when divided by 6. I call these numbers "core numbers." I examined the patterns and properties of core numbers and used programming to test how these patterns appear in many different cases. Through this process, I found that some special odd numbers, created from core numbers, may play an important role in understanding and possibly solving the Collatz conjecture.

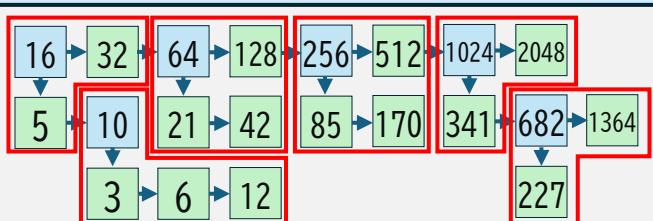
## はじめに

コラッツ予想を知っているだろうか？任意の自然数に対し、Ⓐ偶数なら2で割る、Ⓑ奇数なら3を掛けて1を足す、という操作を繰り返すとどんな数でも1になる、という予想だ。これは現在未解決の問題である。私は、こんなにシンプルな未解決問題があるのか、ということに驚き、興味を持った。

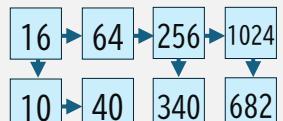
この研究に取り組むにあたり、もちろん解決出来たら最高だが、一学生の限界をわきまえ、**自分なりのやり方で追い詰めていこう**、というのが単純なテーマである。

## 思考

コラッツ予想について考えてみたところ、とある性質を発見した。



1を引いて3で割る(Ⓑの操作の逆)ことができる数(図の青四角)があり、この数をmとする。と、 $m \equiv 4 \pmod{6}$ となる。私はなぜかこの性質に惹かれ、これを「核の数」と呼ぶことにした。また、核の数から別の核の数までの数(図の赤枠で囲まれている数字)を「核の数群」とした。以下、核の数群を表すときその代表となる核の数を用いて示す。



上図のように、この手法を用いることで、表示上の大幅な圧縮ができる。

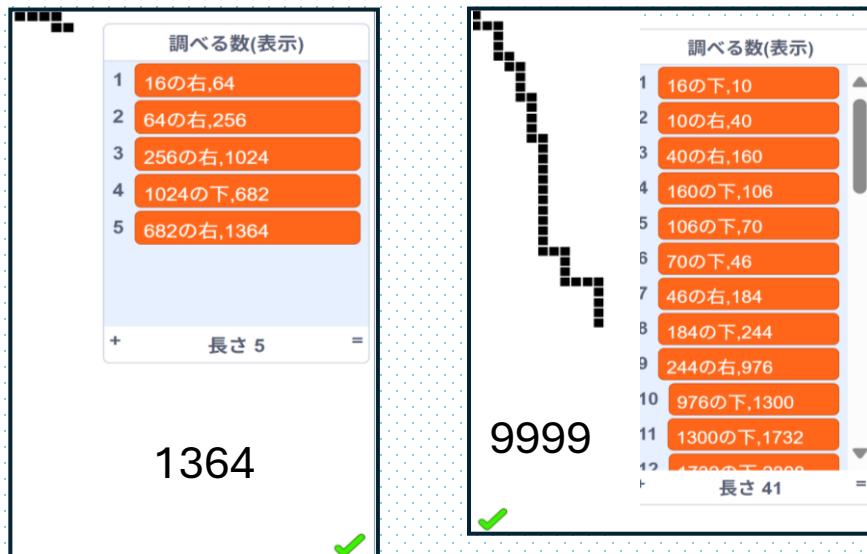
このような考えに基づくと、コラッツ予想の課題は、  
①すべての数がいずれかの核の数群に含まれるか  
②核の数が別な核の数とつながるか  
に集約される。

## プログラミング

ここで、課題①②が成り立つという仮定のもと、任意の自然数に対し、16からの核の数群の道順をナビゲートするプログラム(scratch)を制作した。

(まず、あらかじめ核の数を100000個、また、核の数の下に来る数を同数、リストとしてまとめておく。入力された任意の数aに対し、対象が何の核の数群に含まれるか調べ、戻り値に核の数を入れる関数を実行し、(戻り値、aが核の数の右にあったか下にあったか、a)をリストにまとめる。(戻り値)/2を対象に代入し関数を実行、リスト化、という作業を戻り値が16になるまで繰り返す。その後、リストを読み込んで道順を表示するのである。)

以下はその結果の一例である。



## 結果

様々な数で試行しても**問題なく実行できた**。課題①②について、ある程度の小さい数では成り立つ。

## 結論

この一連の実験で、キーとなるのは、核の数でなく、**核の数の下に手を伸ばす奇数**であることに気付いた。この奇数の性質を明らかにすることがコラッツ予想の根幹にかかるだろう。

作成したプログラムは  
ここから見れる→

